Explaining Educational Recommendations through a Concept-Level Knowledge Visualization

Jordan Barria-Pineda
University of Pittsburgh
Pittsburgh, PA, USA
jab464@pitt.edu

Peter Brusilovsky
University of Pittsburgh
Pittsburgh, PA, USA
peterb@pitt.edu

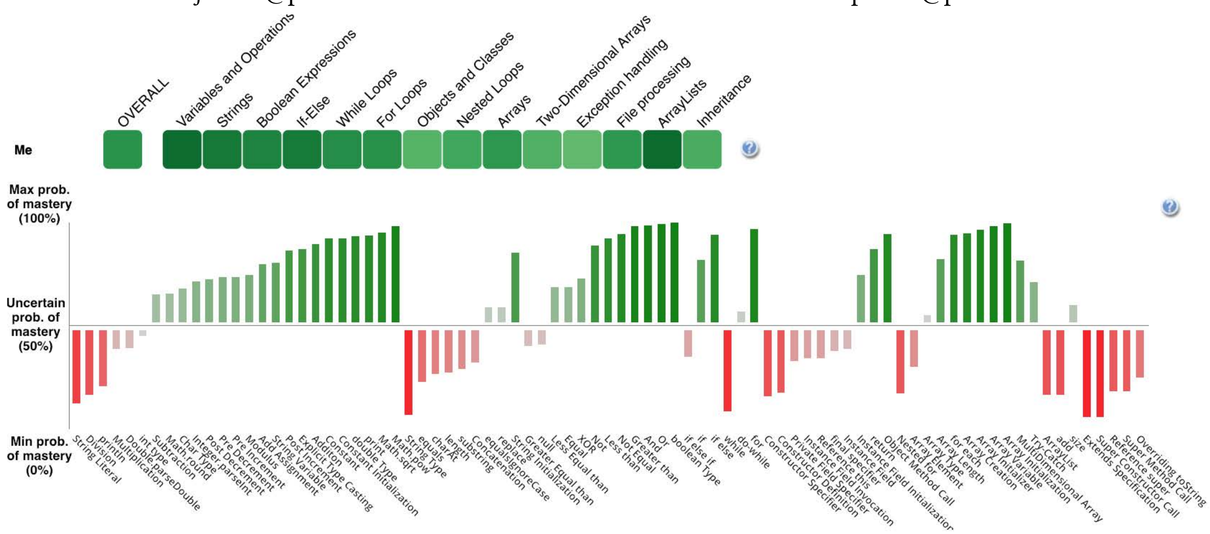

Figure 1: Mastery Grids system showing student progress at topic (upper colored cells) and conceptual level (bar chart)

ABSTRACT

In this demo paper, we present a visual approach for explaining learning content recommendation in the personalized practice system Mastery Grids. The proposed approach uses a concept-level visualization of student knowledge in Java programming to demonstrate why specific practice content is recommended by the system. The visualized student knowledge is estimated by a Bayesian Knowledge Tracing approach, which traces student problem-solving performance. The visual explanatory components, which show both a fine-grained and aggregated knowledge level, are presented to the students along with textual explanations. The goal of this approach is to display the suitability of each recommended item in the context of a student's current knowledge and goal, i.e., the current topic they are studying.

CCS CONCEPTS

• **Applied computing** → **Interactive learning environments**; • **Human-centered computing** → *Information visualization.*

IUI '19 Companion, March 17–20, 2019, Marina del Rey, CA, USA

ACM ISBN 978-1-4503-6673-1/19/03…$15.00
https://doi.org/10.1145/3308557.3308690

KEYWORDS

educational recommendations, explainability, open learner models

ACM Reference Format:

Jordan Barria-Pineda and Peter Brusilovsky. 2019. Explaining Educational Recommendations through a Concept-Level Knowledge Visualization. In *24th International Conference on Intelligent User Interfaces (IUI '19 Companion), March 17–20, 2019, Marina del Rey, CA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3308557.3308690

1 INTRODUCTION

Over the past few years, the research on Explanations for Recommender Systems attracted attention of many researchers along with the broader trend of Explainable AI/Machine Learning. These efforts aim on helping recommender system users to understand why a specific item or a certain decision is being recommended. Recommendations' explanations have been studied in many scenarios, such as e-commerce or people and location recommender systems [5]. However, little work has been done in the context of online educational systems, i.e., exploring how explanations can benefit or hinder the adoption of recommendations in learning scenarios. In fact, [4] argues that explainability is one of the challenges for educational recommender systems and points to information visualization as a possible way to address this issue.

2 EXPLANATIONS AND KNOWLEDGE VISUALIZATION IN E-LEARNING

There is a small body of research on how explanations in recommender systems for learning can improve factors related to student engagement, such as persuasiveness, learning efficiency, satisfaction, etc. [5]. On the other hand, there is a solid body of work on Open Learner Models (OLMs) focused on visualizing student knowledge [2]. In our earlier work [1], we explored a fine-grained visualization of student knowledge, which reflected the distribution of knowledge gained on every programming concept associated with every learning activity in the platform. This visualization helped students understand their knowledge on a deeper level. The work presented below attempts to fill that gap between OLM and educational recommendations. We argue that OLM interfaces could be used to explain learning content recommendations when they are generated based on student level of knowledge of the domain.

3 FINE-GRAINED OLM FOR EXPLAINING EDUCATIONAL RECOMMENDATIONS

In the current demo, we show how concept-level knowledge visualization could be used to augment a learning content recommendation engine, that recommend the most appropriate activities to students to fill their knowledge gaps. To determine these gaps, the level of knowledge on each concept is updated after each problem attempt, based on its result and the bayesian network structure of the domain (concepts-activities) [3].

The main features of our visual explanation interface are:

(1) *Concepts' mastery bar chart*: the estimation of the student mastery of domain concepts is shown through a simple bar chart (see Figure 1). In order to emphasize when the student model is more or less confident about the student mastery on a concept, we use 50% mastery probability as the zero of the y-axis. This percentage reflects that the model is not sure about student mastery (or lack of it), and it is also used to initialize the probabilities per concept in the *cold-start* scenario (no observable evidence of student learning). Accordingly, whenever the student shows evidence that s/he is learning a concept, the mastery value increases above this base probability and hence the corresponding concept bar increases its length towards the positive y-axis. In contrast, if the learner starts failing i.e. giving evidence that s/he is having troubles in learning an specific concept, the estimated mastery probability decreases below this base value. We reflect this through an increase in the corresponding concept bar length, but in this case, towards the negative part of the y-axis. We encode the bars' color following the same rule: when the mastery probability is above 50%, we use green and it gets darker when closer to 100%, whereas when below 50% we use red and it gets more intense when it is closer to 0%. Further, in order to give more context about the concepts that the student should set as her/his study goal, the "focus concepts" for the current topic are highlighted with a dashed frame (see Figure 2). It is important to mention that this visualization component can be used regardless of the student modeling approach used for estimating student knowledge level, as the visualization only uses the mastery estimates' values.

(2) *Recommendation gauge*: The score that represents the suitability of certain learning content given its conceptual composition is shown through a gauge. When a learning activity is *mouseovered*, one of three gauge segments will be targeted by the needle (see Figure 2) according to its appropriateness to her/his level of knowledge. The three gauge categories are the following: (a) *Too hard*: if the the estimated probability of a successful attempt is too low (red color), (b) *Learning opportunity*: activities in which some concepts are not mastered yet, but some important ones are mastered that students can build on their knowledge (green color), and (c) *Too easy*: content that will not report any important learning increase given that the underlying concepts are already mastered (gray color).

(3) *Textual explanation*: a text describing the recommendation rule triggered for each suggested item is shown whenever these activity cells are *mouseovered* (see Figure 2). Although for this demo we use a rule-based recommendation algorithm [3], the more the recommender engine weights the level of knowledge for its calculations, the better for increasing student understanding of this process.

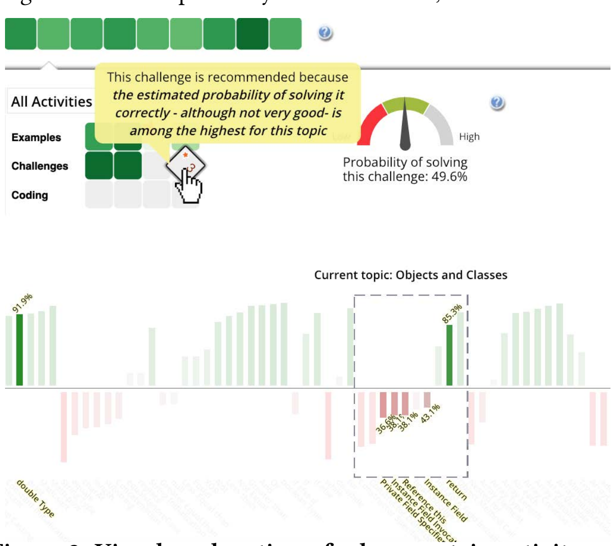

Figure 2: Visual explanation of why a certain activity was recommended to the student, conformed by the concept bar chart, a *recommendation gauge* and a textual explanation.

4 FUTURE WORK

We plan to evaluate this interface in a controlled experiment by using an eye-tracking setup in order to study how students explore and use the different explanatory components for making decisions.

ACKNOWLEDGMENTS

This work was funded by CONICYT PFCHA/Doctorado Becas Chile/2018 - 72190680.

REFERENCES

[1] Jordan Barria-Pineda, Julio Guerra-Hollstein, and Peter Brusilovsky. 2018. A Fine-Grained Open Learner Model for an Introductory Programming Course. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. 53–61.

[2] Susan Bull and Judy Kay. 2007. Student Models That Invite the Learner In: The SMILI:() Open Learner Modelling Framework. *Int. Journal of Artificial Intelligence in Education* 17, 2 (2007), 89–120.

[3] Roya Hosseini. 2018. *Program Construction Examples in Computer Science Education: From Static to Adaptive Engaging Learning Technology.* Ph.D. Dissertation. University of Pittsburgh.

[4] Nikos Manouselis, Hendrik Drachsler, Katrien Verbert, and Erik Duval. 2012. *Recommender Systems for Learning.* Springer, New York, NY, USA.

[5] Nava Tintarev and Judith Masthoff. 2011. Designing and Evaluating Explanations for Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor (Eds.). Springer, Boston, MA, USA, 479–510.